

Slovenská technická univerzita v Bratislave  
Fakulta informatiky a informačných technológií

## **Herný ovládač**

Semestrálny projekt

Meno: Bc. Mário Csaplár

Študijný program: Softvérové inžinierstvo

Ročník: 1.

Predmet: Vnorené systémy

Akademický rok: 2015/2016

# Obsah

<b>1</b>	<b>Analýza</b>	<b>1</b>
1.1	Zadanie projektu . . . . .	1
1.2	Riadiace zariadenie . . . . .	1
1.2.1	Mikroprocesory rodiny Intel . . . . .	1
1.2.2	Prototypové dosky Arduino . . . . .	2
1.3	Tlačidlá . . . . .	4
1.4	Joystick . . . . .	6
1.5	Nadstavbová doska . . . . .	6
1.6	Vývojové prostredie . . . . .	7
1.7	Cieľové prostredie . . . . .	8
<b>2</b>	<b>Opis riešenia</b>	<b>9</b>
2.1	Špecifikácia požiadaviek . . . . .	9
2.2	Návrh zapojenia . . . . .	10
2.3	Návrh firmvéru . . . . .	10
2.4	Testovanie . . . . .	12
<b>3</b>	<b>Zhodnotenie výsledkov</b>	<b>13</b>
<b>4</b>	<b>Technická dokumentácia</b>	<b>14</b>
4.1	Použité knižnice . . . . .	14
4.2	Piny nadstavbovej dosky . . . . .	14
4.3	Inicializácia . . . . .	15
4.4	Hlavná slučka . . . . .	15
	<b>Literatúra</b>	<b>17</b>

# 1 Analýza

Táto kapitola obsahuje analýzu riadiacich zariadení a periférií, ktoré môžu byť použité na realizáciu vnoreného systému určeného zadaním v sekcii 1.1. V úvodných sekciách sú opísané hardvérové prostriedky, po ktorých nasleduje opis vývojového a cieľového prostredia.

## 1.1 Zadanie projektu

Navrhňte a zrealizujte vnorený systém založený na platforme Arduino, ktorý bude vystupovať ako kombinovaná USB klávesnica a myš. Vonkajšie vstupy sa budú skladať z analógových joystickov slúžiacich na presné určenie smeru pohybu a niekoľkých tlačidiel, ktoré po stlačení rozvinú vlastné makrá, prípadne upraví presnosť pohybu. Voľbu makier a zapojenie prispôbte tak, aby sa pomocou navrhnutého vnoreného systému dala ovládať vybraná jednoduchá hra.

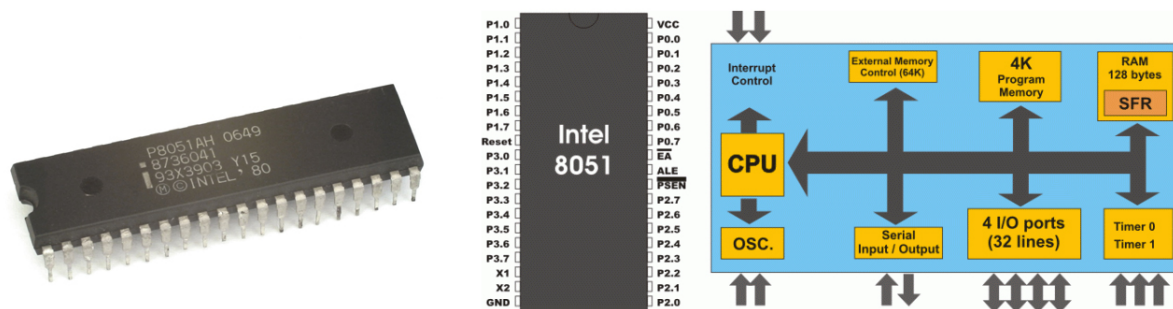
## 1.2 Riadiace zariadenie

Výber riadiaceho člena bol prvý krok pri tvorbe špecifikácie zadania. Bolo potrebné zistiť, aké technické prostriedky sú v súčasnosti dostupné na trhu, posúdiť ich z viacerých hľadísk a jedno zariadenie si vybrať.

### 1.2.1 Mikroprocesory rodiny Intel

Predmet *Mikropočítače* priniesol praktický pohľad na prácu s procesormi rodiny Intel x51 a Intel x86. Procesor 8051 znázornený na obrázku 1.1 je osembitový mikropočítač s Harvardskou architektúrou, teda s oddelenou pamäťou inštrukcií a pamäťou programu.

Dátová zbernica je 8-bitová, adresová zbernica 16-bitová, čo umožňuje 64 kB priestoru pre pamäť. Podporuje sériovú komunikáciu prostredníctvom UART [3]. Nevýhodou je náročná inicializácia, nutnosť externého napájania, zdĺhavé programovanie prostredníctvom programátorov a nerobustné zapojenie periférií cez *breadboard*.



Obr. 1.1: Vľavo fotografia procesora rodiny x51, vpravo blokové schémy.

## 1.2.2 Prototypové dosky Arduino

Arduino je open-source platforma založená na elektronickej doske s jednočipovým mikro počítačom, ku ktorému sú pripojené voliteľné senzory fyzického prostredia. Využitelná je najmä v prvotných fázach realizácie projektov, kedy je potrebné overiť si plánovanú funkcionálnosť či technologické limity senzorov. Zároveň je vhodná aj ako cieľová platforma menších projektov, medzi ktoré sa radí aj tento vnorený systém.

Výhodami tejto platformy sú:

- Cenová nenáročnosť. Najlacnejšie verzie balíčkov Arduino sa pohybujú v cenovej relácii od 10 € do 50 €.
- Vysoká variabilita meracích modulov. K dispozícii sú takmer všetky druhy senzorov, vrátane teplotného, tlakového, záťažového či zvukového.
- Nezávislosť od platformy. Aplikácie Arduino sú kompatibilné so všetkými bežnými operačnými systémami.
- Vlastné programovacie prostredie. Prostredie založené na objektovo-orientovanom jazyku C++ umožňuje vysokoúrovňové riadenie dokonca aj analógových častí systému.

Na trhu existuje viacero variantov dosiek Arduino, pričom všetky sa programujú prostredníctvom USB. Technologicky sa však rozdeľujú na dosky so samostatným programovacím a vykonávacím mikroprocesorom a dosky s jediným mikroprocesorom.

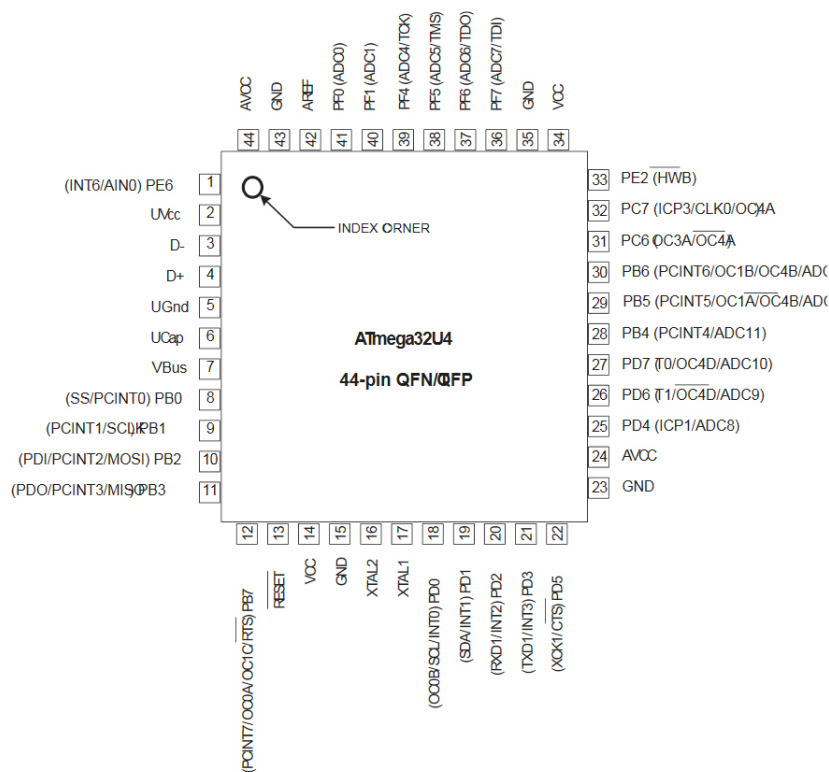
Základom väčšiny dosiek je rodina procesorov ATmega od spoločnosti Atmel. Pre tento projekt je vhodný práve variant dosky s jediným procesorom, pretože vtedy je možné dosku nielenže programovať cez USB, dokáže súčasne vystupovať ako samostatné USB zariadenie a sériovo komunikovať s nadradeným systémom.

Ďalšou výhodou platformy Arduino je rozšíriteľnosť o predpripravené súbory funkcionality vo forme nadstavbovej dosky (angl. *shield*). Typickým príkladom je nadstavbová doska

s meracími členmi, na ktorú je napojená ďalšia nastavbová doska s LED displejom. Keďže takmer všetok hardvér rodiny Arduino má jednotné rozmery, je zaručená vysoká kompatibilita a dokonca možnosť vrstvenia nastavbových dosiek.

Doska *Arduino Leonardo* bola uvedená na trh v roku 2012 a hoci už dnes (máj 2016) nie je oficiálne podporovaná, naďalej je dostupná v predaji. Zakladá sa na jedinom mikrokontroléri ATmega32u4, ktorého schéma je uvedená na obrázku 1.2 [2].

Tento 8-bitový mikrokontrolér disponuje 44 pinmi (najviac 26 vstupno-výstupných) pri klasických hodnotách napätia TTL a pracovnej frekvencii 16 MHz. K dispozícii je 2,5 kB SRAM, 1 kB EEPROM a rozhranie JTAG.

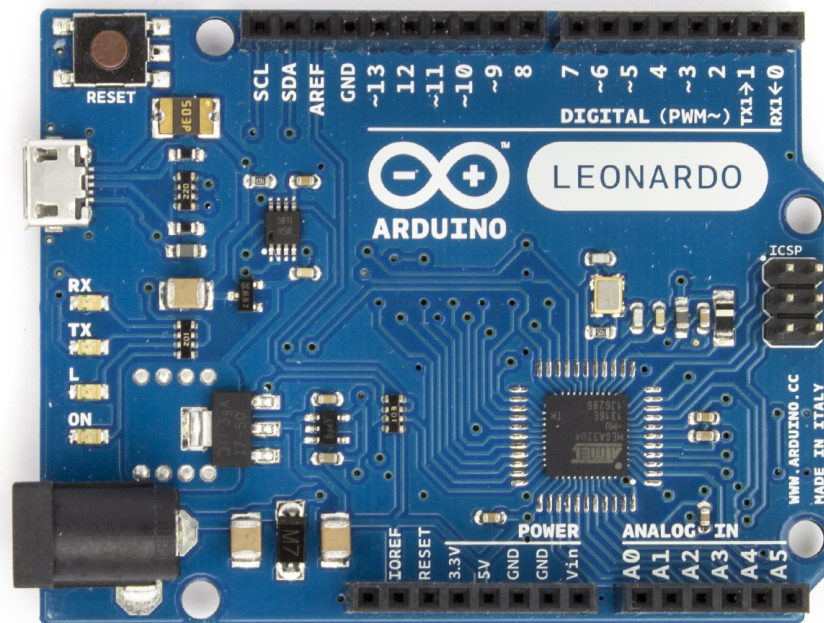


Obr. 1.2: Schéma pinov mikrokontroléru Atmel ATmega32u4.

Samotná doska Arduino Leonardo je znázornená na obrázku 1.3. Fyzické rozmery sú 68,6 mm x 53,3 mm s váhou 20 g. Disponuje 32 kB pamäťou typu Flash, z ktorej 4 kB zaberá bootloader. Pre používateľa vystavuje 20 digitálnych vstupno-výstupných portov, z čoho 12 je možné použiť na zapojenie analógových prvkov a 6 ako výstupy s pulznou šírkovou moduláciou [1].

Hoci je pracovné napätie 5V, odporúčané vstupné napätie sa nachádza v rozpätí 7V až 12V, pričom je zakázané presiahnuť 20V. Hodnota jednosmerného prúdu na vstupno-

výstupných pinoch sa môže pohybovať okolo 40 mA.



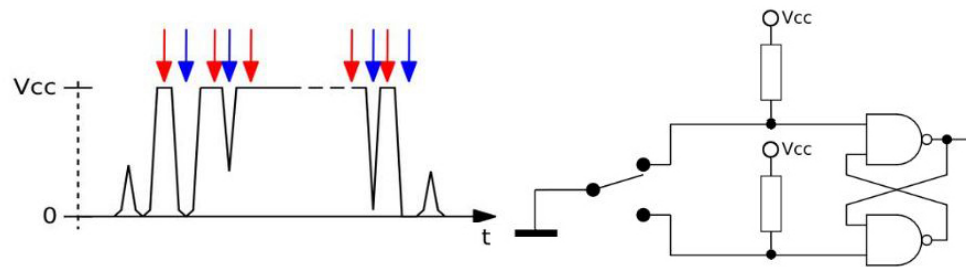
Obr. 1.3: Fotografia prototypovej dosky Arduino Leonardo.

Najväčšou výhodou tejto dosky je už spomínaný jediný mikrokontrolér Atmel ATmega32u4, ktorý slúži na programovanie cez mikroUSB a zároveň ako člen vykonávajúci používateľský program. Vďaka tomu môže Arduino Leonardo vystupovať ako USB zariadenie a prostredníctvom sériovej komunikácie emulovať klávesnicu, myš alebo oboje.

## 1.3 Tlačidlá

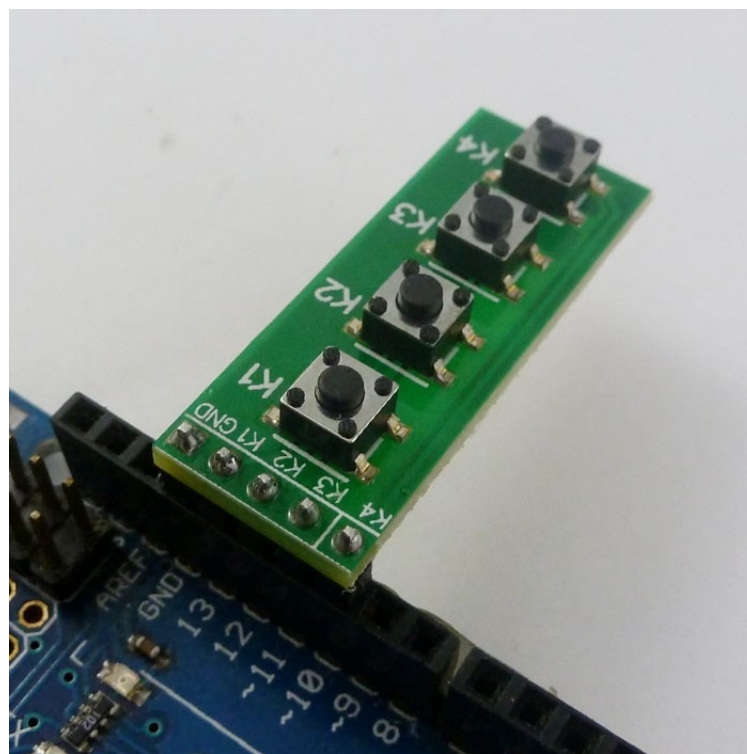
Tlačidlo je jedným zo základných elektronických prvkov slúžiacich na získavanie vstupu od používateľa. Technicky je realizované ako spínač, avšak jeho konštrukcia si vyžaduje ošetrovanie proti zákmitom. Zákmity sú opakované krátkodobé, mikroskopické spojenia a odpojenia vodičov, ktoré nastávajú pri stlačení a uvoľnení tlačidla. Ošetrojú sa typicky preklápacím obvodom PO-SR, ako je to znázornené na obrázku 1.4.

Ako bolo spomenuté, platforma Arduino umožňuje pripojenie externých digitálnych alebo analógových zdrojov signálu. Existuje preto vysoká variabilita zapojenia tlačidiel, kam sa radí ošetrovanie proti zákmitom štandardných tlačidiel na breadboarde, zapojenie už ošetrovaných tlačidiel alebo nadstavbová doska obsahujúca n-ticu tlačidiel na obrázku 1.5.



Obr. 1.4: Vľavo mikroskopické rozpojenia a spojenia kontaktov, vpravo schéma ošetrenia.

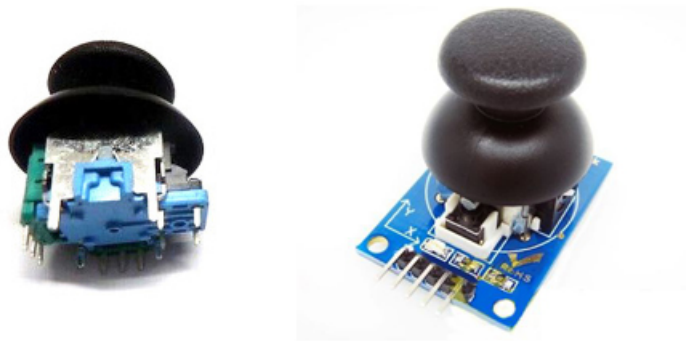
Keďže účelom tlačidiel je pomôcť používateľom ovládať jednoduchú hru, ako najvhodnejšie riešenie sa javí vyhotovenie na kompaktnej, fyzicky robustnej nadstavbovej doske. Z hľadiska systému je dôležité uvedomenie, že používateľ by s ním v kontexte hrania nezaobchádzal jemne. Štandardné napájanie všetkých spomínaných riešení je 5V, výstupom tlačidla je buď logická 0 alebo logická 1 v rámci úrovni TTL.



Obr. 1.5: N-tica tlačidiel na nadstavbovej doske.

## 1.4 Joystick

Joystick je analógový prvok skladajúci sa z dvoch potenciometrov, ktoré indikujú náklon v osiach X a Y. Vďaka tomu sa dá získať pomerne dobrá informácia o tom, akým spôsobom používateľ joystick nakláňa, a to aj v prípade, keď ide o náklon do oboch smerov súčasne.



Obr. 1.6: Vľavo joystick pripojiteľný na breadboard, vpravo joystick ako nadstavba Arduina.

Opäť existuje viacero riešení z pohľadu zapojenia, pričom zoznam je takmer identický ako v prípade tlačidla. Je možné zapojiť samostatný potenciometer na breadboard alebo zakúpiť už hotové riešenie vo forme nadstavbovej dosky pre platformu Arduino. Obe vyhotovenia graficky porovnáva obrázok 1.6.

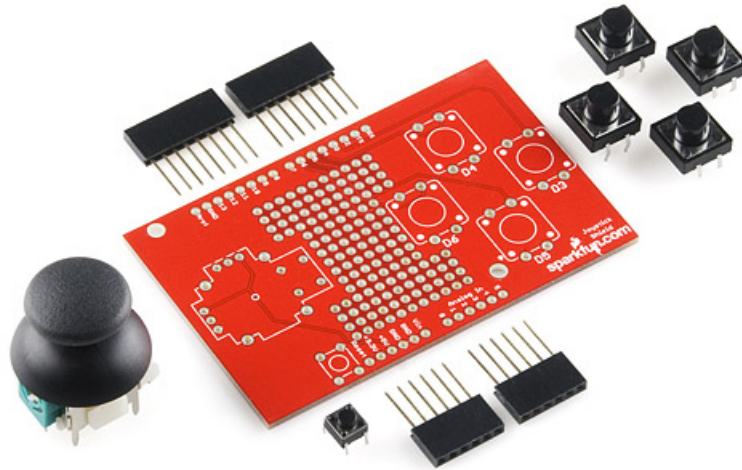
## 1.5 Nadstavbová doska

Pre platformu Arduino je k dispozícii široká paleta nadstavbových dosiek, z ktorých niektoré boli predmetom predošlých sekcií analýzy. Namiesto použitia dvoch samostatných nadstavbových dosiek pre n-ticu tlačidiel a joystick je možné použiť jediné vyhotovenie pod názvom *joystick shield* [4].

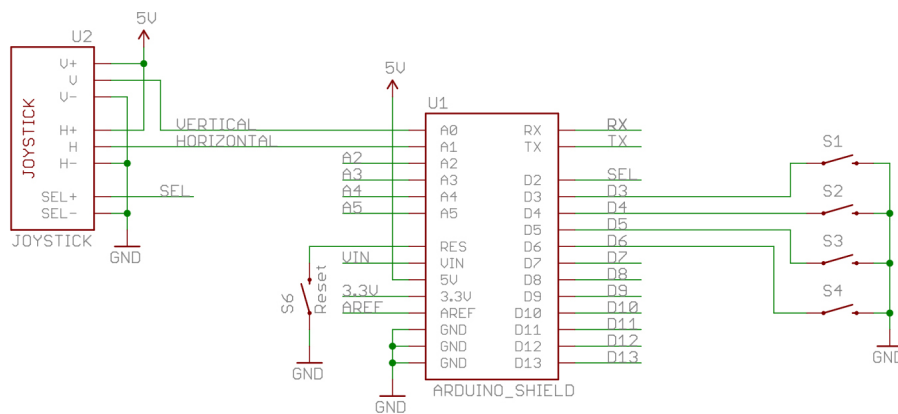
Najdostupnejším predstaviteľom je *SparkFun Joystick Shield Kit* od spoločnosti SparkFun. Obsahuje 6 digitálnych tlačidiel a jeden joystick na robustnom vyhotovení nadstavbovej dosky. Zároveň obsahuje piny na pripojenie ďalšej vrstvy nadstavbovej dosky v prípade potreby. Vyhotovenie je znázornené na obrázku 1.7.

Alternatívou je *Funduino Joystick Shield*, ktorý poskytuje zhodnú funkcionálnosť, iba s odlišným fyzickým vyhotovením. Schéma zapojenia na 1.8 je však zhodná pre oba opísané modely.





Obr. 1.7: SparkFun Joystick Shield Kit pred zložením.

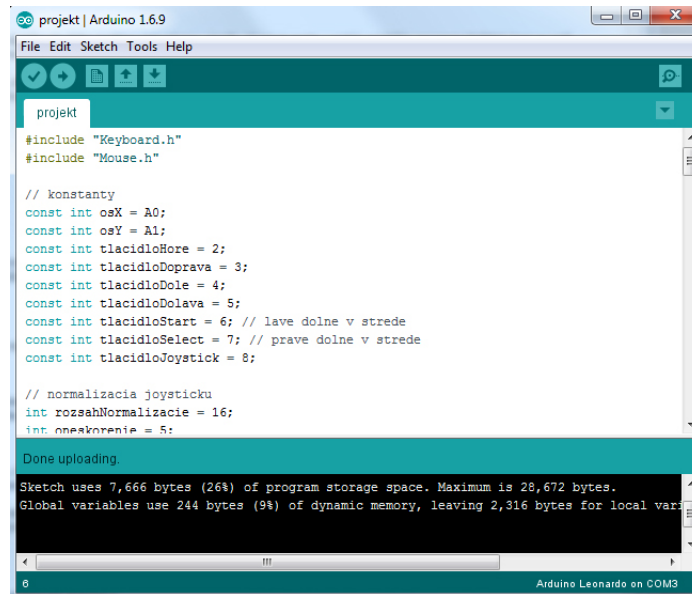


Obr. 1.8: Schéma SparkFun Joystick Shield Kit.

## 1.6 Vývojové prostredie

Vývojové prostredie pre platformu Arduino je Arduino IDE, v ktorom je priamo možné písať a ladiť programy v jazyku príbuznom C. Zároveň je možné overiť integritu pripojenej dosky, zvoliť spôsob programovania, prijímať a posielať dáta cez sériovú linku a nahrávať program do pamäte dosky.

K dispozícii je takisto rozsiahla dokumentácia s množstvom ukážkových riešení. Obrázok s editovaním zdrojového kódu projektu je znázornená na obrázku 1.9. Po každom skompilovaní prostredie uvádza štatistiku využitia dostupnej pamäte hardvérového prostredku.



Obr. 1.9: Arduino IDE.

## 1.7 Cieľové prostredie

Cieľovým prostredím pre tento vnorený systém je osobný počítač alebo notebook s niektorým z rodiny operačných systémov Microsoft Windows. Dôvodom je automatické rozpoznávanie HID (angl. *human interface device*), teda zariadení slúžiacich na interakciu človeka s počítačom. Štandard USB definuje túto triedu zariadení, pričom každé zariadenie sa musí pri pripojení ohlásiť host'ovskému systému.

Hoci systém Windows implicitne nepodporuje dva kurzory myši, nemá žiaden limit pre počet pripojených HID. Ak teda používateľ zapojí viacero klávesníc alebo viacero myší, systém Windows spracúva vstupné signály ako od jediného zariadenia daného typu. Zároveň nelimituje jediný port len na jeden typ zariadenia, pokojne môže jeden USB port slúžiť ako úložné zariadenie, zdroj signálu periférií a rodič ďalších USB zariadení.

## 2 Opis riešenia

Obsahom tejto kapitoly je opis celého postupu riešenia zadaného vnoreného systému. Ako prvé sú v sekcii 2.1 špecifikované požiadavky, ktoré musí systém spĺňať. V nasledujúcej časti 2.2 sa nachádzajú návrhy zapojenia a firmvéru, zatiaľ čo záver kapitoly je venovaný testovaniu.

### 2.1 Špecifikácia požiadaviek

Špecifikácia požiadaviek sa dotýka nárokov na systém, ktorý má byť možné použiť na ovládanie jednoduchej hry. Vychádza z analýzy možností realizácie poskytnutej v predošlej kapitole.

Vnorený systém by mal podporovať viacero režimov činnosti, aby ho bolo možné účinne používať. Základným režimom, v ktorom bude možné overiť funkcionality štandardnej myši a klávesnice, bude režim *stand-by*.

Vybraným tlačidlom na doske by malo byť možné systém prepnúť do režimu plnej funkcionality herného ovládača. O aktuálnom stave by mal systém informovať používateľa prostredníctvom rozsvietenia LED.

Systém by mal navonok pôsobiť ako USB zariadenie, pričom by zohrával rolu USB HID z pohľadu myši aj klávesnice. Na základe analýzy bolo zistené, že je možné plniť obe role súčasne v zapojení do jediného USB portu.

Klávesnica by mala byť simulovaná niekoľkými tlačidlami, ktoré by boli schopné poslať nadradenému systému vybrané znaky. Tieto znaky bude následne možné priradiť v rôznych hrách určitým akciám. Zároveň by mal systém poskytovať niekoľko makier, teda stlačením jedného tlačidla by mal vyslať užitočnú kombináciu klávesov.

Funkcionalita myši by mala byť napodobnená joystickom, pričom by systém mal podporovať viacero rýchlostí ovládania kurzora. Stlačením joysticku by malo byť možné vyslať signál stlačenia ľavého tlačidla myši.

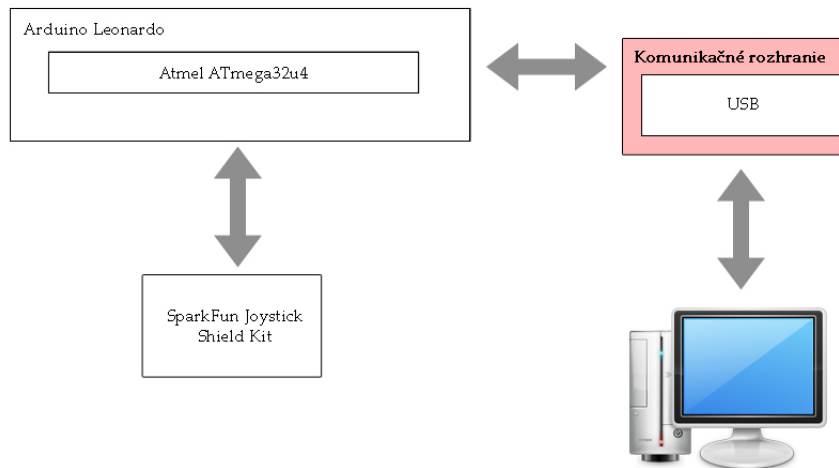
Z pohľadu fyzického vyhotovenia by mal byť vnorený systém dostatočne robustný, aby sa pri neopatrnom zaobchádzaní nepoškodil. Ide o klasickú požiadavku na väčšinu herných

ovládačov, ktoré v napätých herných situáciách musia odolať fyzickému tlaku vyvíjanému používateľom.

## 2.2 Návrh zapojenia

Vďaka realizácii prostredníctvom nadstavbovej dosky *Funduino Joystick Shield* je bloková architektúra systému veľmi jednoduchá. Obrázok 2.1 znázorňuje tri konceptuálne časti systému. Základnom je prototypová doska Arduino Leonardo, na ktorej je umiestnená nadstavbová doska Funduino Joystick Shield.

Prostredníctvom rozhrania mikroUSB je prototypová doska napájaná z nadradeného počítača, s ktorým zároveň sériovo komunikuje. Keďže nadstavbová doska využíva všetky piny prototypovej dosky, mapovanie pinov je 1:1 voči referenčnej dokumentácii.

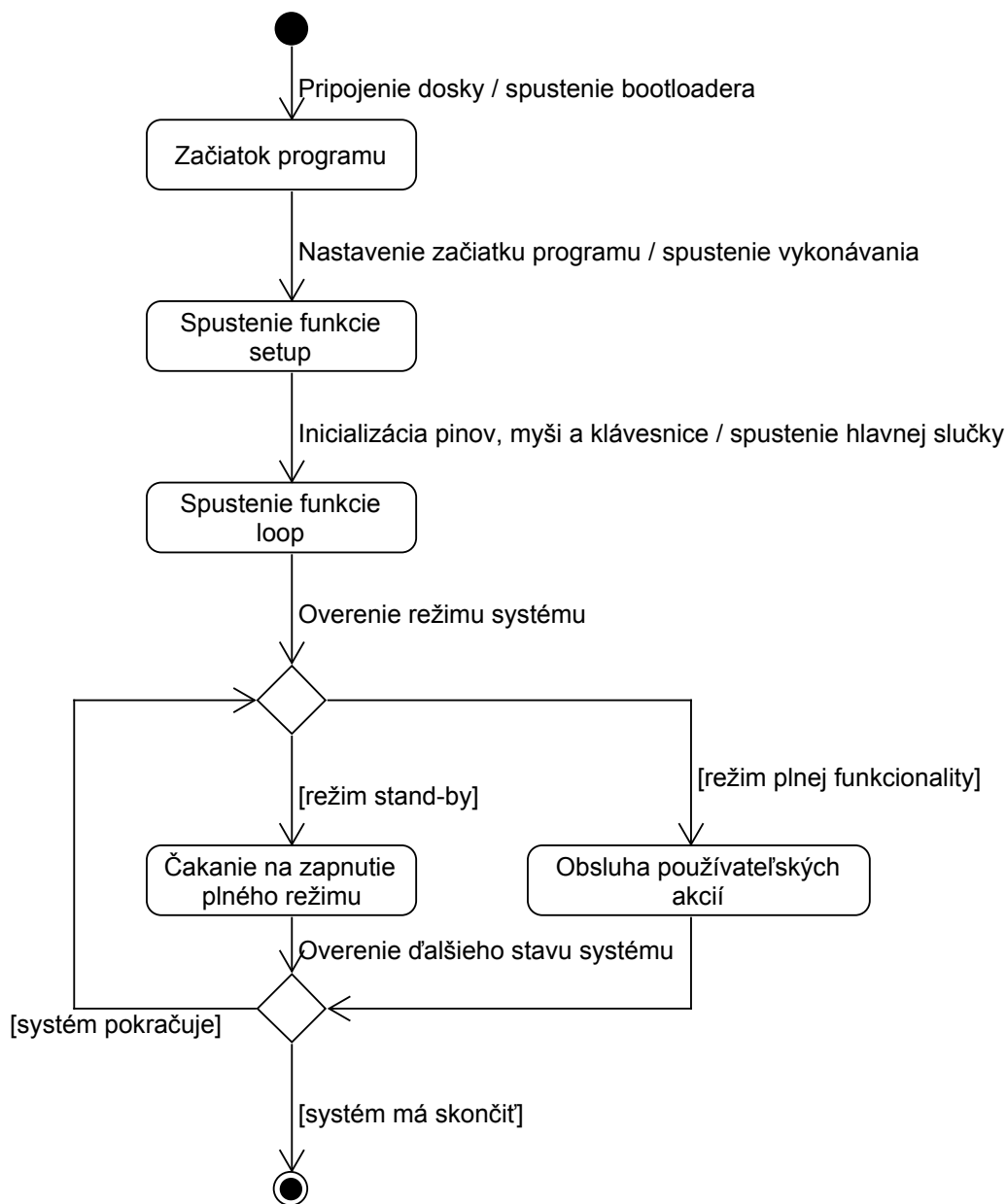


Obr. 2.1: Bloková schéma zapojenia.

## 2.3 Návrh firmvéru

Návrh firmvéru zodpovedá špecifikácii požiadaviek na systém zo sekcie 2.1, pričom hlavný tok programu je znázornený na diagrame činností 2.3. Po zapojení systému sa spustí funkcia *setup()*, v ktorej je vykonaná celá inicializácia. Následne sa v hlavnej slučke programu kontroluje, či je systém v režime *stand-by* alebo v plnom režime funkcionality.

V režime *stand-by* umožňuje stlačením tlačidla *Select* vojsť do plného režimu činnosti a z dôvodov testovania umožňuje stlačením tlačidla *Štart* rozvinúť makro *CTRL+ALT+DEL*, ktoré bolo v procese tvorby systému často využívané.



Obr. 2.2: Diagram činností pre firmvér.

V režime plnej funkcionality sa najprv odčítava hladina analógových signálov z potenciometrov, ktorá sa normalizuje, aby pohyb myšou bol plynulý. Normalizácia je modifikovaná prostredníctvom koeficientu rýchlosti, ktorý sa dá prepínať tlačidlom *Štart* medzi tromi rýchlostnými hodnotami.

Následne sa odčítavajú hodnoty štyroch „klávesových“ tlačidiel, ktoré v prípade aktívnej hodnoty vyšlú nadradenému systému príslušný kláves „w“, „a“, „s“ alebo „d“ zodpovedajúce

rozložení klávesov na klávesnici.

## 2.4 Testovanie

Navrhnutý systém bol testovaný priebežne počas celého procesu tvorby, najmä s dôrazom na overovanie riadiacej logiky. Na overenie priradenia pinov na nadstavbovej doske voči referenčnej dokumentácii bol vytvorený jednoduchý program, ktorý odčítaval hodnoty všetkých digitálnych aj analógových vstupov a manuálnym testovaním boli menené hodnoty.

Po overení riadiacej logiky bola testovaná funkcionálna myši. Boli odskúšané všetky tri režimy rýchlosti pohybu aj s funkcionálnou stlačenia ľavého tlačidla myši. Na základe testovania bolo možné usúdiť, že zvolené rýchlosti sú pri nezmenenej konfigurácii myši v systéme Windows vhodne zvolené a je pomocou nich možné kliknúť, napríklad, aj na odkazy na webových stránkach.

Na otestovanie vhodnosti celého zariadenia bola vybraná hra *Crash Bandicoot 3: Warped*, ktorá v roku 1999 vyšla pre hernú konzolu PlayStation. Tú je možné emulovať pod systémom Windows, napríklad, prostredníctvom emulátora *ePSXe*. Táto hra bola vybraná najmä z dôvodu, že pôvodný joystick *DualShock* pre konzolu PlayStation je vlastnosťami podobný navrhnutému systému.

Emulátor bol spustený pod dvoma systémovými konfiguráciami, kde bola zistená dôležitá skutočnosť. Pri prenose vnoreného systému medzi počítačmi je potrebné podľa výkonu upraviť konštantu *delay*, ktorá je štandardne nastavená na 5 milisekúnd po každom ukončení iterácie hlavnej slučky.

Pri testovaní na notebooku s jednojadrovým procesorom na frekvencii *1,3 GHz*, *256 MB RAM* a OS Windows XP bolo toto oneskorenie nutné úplne odstrániť, pretože odozva systému bola príliš pomalá. Po odstránení oneskorenia bolo možné ovládať hru aj na tejto konfigurácii bez výrazných problémov.



Obr. 2.3: Obrázok z hrania Crash Bandicoot 3: Warped v emulátore ePSXe.

## 3 Zhodnotenie výsledkov

Cieľom tohto semestrálneho zadania bolo vytvoriť vnorený systém, ktorý by realizoval zvolenú špecializovanú funkcionálnu. Tou sa v tomto prípade stala možnosť ovládať pomocou vybraného hardvérového zapojenia jednoduchú hru.

Analýza sa venuje dostupným možnostiam realizácie riadiaceho zariadenia, digitálnych členov a analógového člena - joysticku. Celé riešenie bolo napokon cielené do platformy Arduino, ktorá vďaka jednoduchosti tvorby firmvéru, kvalitnej dokumentácii a vysokej variabilite rozširujúcich členov predstavuje moderný prototypovací systém.

Namiesto vlastnej realizácie zapojenia vonkajších členov bola vybraná špecializovaná nadstavbová doska, ktorá poskytuje pre potreby herného ovládača dostatočnú fyzickú robustnosť. Týmto krokom sa zároveň zjednodušila komplexnosť zapojenia a eliminovalo sa riziko vlastných chýb, ktoré mohli vzniknúť pri tvorbe takéhoto zapojenia.

V nasledujúcom kroku bol vytvorený firmvér pre vnorený systém, ktorý umožňuje zariadeniu fungovať v dvoch režimoch. V režime *stand-by* nemá ako kombinované USB zariadenie žiaden vplyv na pozíciu kurzora myši, ale podporuje vysielanie jednoduchého makra. V plnom funkcionálnom režime podporuje pohyb kurzora myši v troch rýchlostných režimoch a vysielanie 4 vybraných klávesov, ktoré je možné namapovať na akcie v hrách.

Testovaním vnoreného systému na dvoch rôznych počítačových konfiguráciách bolo potvrdené, že výsledné zapojenie je v súlade s princípom *plug & play*. Systém Windows si dokáže automaticky nájsť potrebné ovládače a uviesť vnorený systém do prevádzky. V tom momente je zariadenie identifikované jednak ako programovateľný obvod Arduino Leonardo, ale zároveň ako kombinovaný USB HID.

Funkcionalita bola otestovaná jednak z pohľadu klasickej myši, kedy sa osvedčili tri rýchlostné režimy na zvýšenie prenosnosti pohybu, najmä ak bolo potrebné kliknúť na odkaz či otvoriť súbor. Následne bola otestovaná samotná funkcionálna systém ako herného ovládača, kedy sa úspešne podarilo systém použiť na ovládanie vybranej hry pre konzolu *PlayStation*. Ako pohyb kurzora myši, tak aj ovládanie hry boli prostredníctvom navrhnutého ovládača subjektívne pohodlné a plynulé.

## 4 Technická dokumentácia

Táto kapitola obsahuje technické detaily implementácie. Na začiatku sú definované knižnice, ktoré boli použité pri implementácii firmvéru v prostredí Arduino IDE. Zvyšok kapitoly sa venuje vybraným častiam programového kódu firmvéru.

### 4.1 Použité knižnice

V súbore *projekt.ino* boli použité vstavané knižnice Arduino IDE *Keyboard.h* a *Mouse.h*. Obe obsahujú metódy *begin()*, ktoré zahajujú komunikáciu myši, respektíve klávesnice s nadradeným systémom.

Pre klávesnicu boli použité metódy *press(char key)* na podržanie klávesu, *release(char key)* na pustenie klávesu a *press(char key)* na vyslanie jediného klávesového impulzu. Pre myš bola použitá metóda *move(int x, int y, int wheel)* posúvajúca kurzor, respektíve koliesko myš v oboch smeroch.

### 4.2 Piny nadstavbovej dosky

V tejto sekcii je uvedený kód, ktorý bol výsledkom analýzy funkcionality nadstavbovej dosky. Bolo zistené, že nie všetky piny zodpovedajú opisu v technickej dokumentácii, dokonca niektoré tlačidlá sú pripojené na analógový aj digitálny pin.

```
const int osX = A0;
const int osY = A1;
const int tlacidloHore = 2;
const int tlacidloDoprava = 3;
const int tlacidloDole = 4;
const int tlacidloDolava = 5;
const int tlacidloStart = 6;
const int tlacidloSelect = 7;
```



```
const int tlacidloJoystick = 8;
```

## 4.3 Inicializácia

Nasledujúci kód sa používa na inicializáciu pinov na vstupné, čo spôsobí nastavenie referenčnej logickej hodnoty, ktorou je v prípade tejto obvodovej dosky logická jednotka. To znamená, že po stlačení tlačidla je aktívnou hodnotou logická nula. Dá sa teda povedať, že všetky tlačidlové vstupy sú negované.

```
void setup() {
  pinMode(tlacidloHore , INPUT);
  pinMode(tlacidloDoprava , INPUT);
  pinMode(tlacidloDole , INPUT);
  pinMode(tlacidloDolava , INPUT);
  pinMode(tlacidloSelect , INPUT);
  pinMode(tlacidloStart , INPUT);
  pinMode(tlacidloJoystick , INPUT);
  Mouse.begin();
  Keyboard.begin();
  Serial.begin(9600);
}
```

## 4.4 Hlavná slučka

Po jednorazovom vykonaní inicializácie sa spustí hlavná slučka, ktorá sa vykonáva až do odpojenia systému. Vybraná časť kódu demonštruje základ riadiacej logiky vzhľadom na dva režimy činnosti systému.

```
void loop() {
  int aktualnyStav = digitalRead(tlacidloSelect);
  if (aktualnyStav != poslednyStav) {
    if (aktualnyStav == LOW) {
      systemAktivny = !systemAktivny;
    }
  }

  poslednyStav = aktualnyStav;
}
```

```
int hladinaX = normalizujOs(osX);  
int hladinaY = normalizujOs(osY);  
  
if (systemAktivny) {  
    Mouse.move(hladinaX, -hladinaY, 0);  
}  
}
```

# Literatúra

- [1] Arduino. Arduino leonardo schematic. Online dostupné na [https://www.arduino.cc/en/uploads/Main/arduino-leonardo-schematic\\_3b.pdf](https://www.arduino.cc/en/uploads/Main/arduino-leonardo-schematic_3b.pdf) (16. máj 2016).
- [2] Atmel. Atmega32u4 datasheet. Online dostupné na [http://www.atmel.com/Images/Atmel-7766-8-bit-AVR-ATmega16U4-32U4\\_Datasheet.pdf](http://www.atmel.com/Images/Atmel-7766-8-bit-AVR-ATmega16U4-32U4_Datasheet.pdf) (16. máj 2016).
- [3] Intel. 8051 datasheet. Online dostupné na <http://pdf1.alldatasheet.com/datasheet-pdf/view/107780/INTEL/8051.html> (16. máj 2016).
- [4] SparkFun. Joystick shield datasheet. Online dostupné na [http://cdn.sparkfun.com/datasheets/Dev/Arduino/Shields/Joystick\\_Shield-v14.pdf](http://cdn.sparkfun.com/datasheets/Dev/Arduino/Shields/Joystick_Shield-v14.pdf) (16. máj 2016).